A Scalable Application for Automatic Internationalization of ISO19139 Metadata in DRDSI[*]

Alberto Gemelli

(personal research)

albertogemelli@hotmail.com

Abstract

Internationalization is the process of transforming software or digital documents so that they can be accessed automatically in different languages in different countries. In this work, a prototype software application named MT@EC-Wrapper, which internationalizes a corpus of XML documents, transforming them into a multilingual parallel corpus, is implemented within the Danube Reference Data Services and Infrastructure (DRDSI) of the EU's INSPIRE project. The application integrates document automation technology with the European Commission's on-line machine translation service MT@EC. This case study achieves the goal of fully automating the transformation process of the DRDSI ISO19139 XML repository into a parallel corpus in the nine official languages of the DRDSI project. The design of this application also addresses the issues of processing performance, control and scalability. The project is compared with similar systems used within the EU institutions; the focus of the analysis is on metadata standards for internationalization and metadata processors.

Keywords: Internationalization, Parallel Corpus, XML, RESTful Service, Microservices, DRDSI, INSPIRE, Document Automation.

DOI: 10.2902/1725-0463.2017.12.art6

1. INTRODUCTION

INSPIRE[1] is a European Union initiative which aims to make geographic information accessible and interoperable for use in studies involving sustainable development. In the context of INSPIRE, the objective of the ongoing *Danube Reference Data Services and Infrastructure* (DRDSI[2]) project is to engage end-users, such as institutional decision makers, Danube region stakeholders, data users and data providers, in the sharing of geospatial information relevant to the EU Strategy for the Danube Region. DRDSI is a web-based platform which provides interfaces for the discovery, visualization and downloading of geospatial datasets. It regularly performs a collection (harvesting) of metadata from remote catalogues available through the OGC *Catalogue Service for the Web* (CSW), and stores them in its local repository.

In this scenario, there is a need to make the DRDSI descriptive metadata content available in multiple languages, and to automate this internationalization process. The internationalization of a metadata XML document is a complex process which includes the extraction of textual content from the document, the execution of translation into numerous languages, and finally the reintegration of these translations into a new metadata document that respects widely recognized standards for international document formatting. If this process is to be repeated for the thousands of documents populating a repository such as DRDSI, it must be automated. The application presented in this paper, named MT@EC-Wrapper (hereafter abbreviated to Wrapper), has therefore been developed so that it can:

- *Internationalize the DRDSI data catalogue*: The data resources harvested by DRDSI are multilingual, since they have been contributed by institutions from different countries, and the accompanying descriptive metadata are given in local languages. Access to metadata will be facilitated for all stakeholders by offering access to the catalogues in their own languages.

- *Accomplish specific requirements of the INSPIRE Directive*: Specific requirements and recommendations regarding the multilinguality of *Spatial Data Infrastructures (SDI)* were expressed in the INSPIRE Directive (INSPIRE, 2013). The multilinguality of the catalogues and information about geographic data was defined as crucial for INSPIRE. Since translating every resource into a common language was not considered as a solution, it was recommended that internationalization should be carried out.

[1] http://inspire.ec.europa.eu/
[2] http://drdsi.jrc.ec.europa.eu/

- *Automatize the transformation process*. The automation of internationalization creates value for the organizations involved, since it makes access to resources more efficient and customizes them, allows automatic updating of already internationalized resources, speeds up the delivery of the translations and creates cost savings.

2. THE INTERNATIONALIZATION PROCESS AND TECHNOLOGY BACKGROUND

2.1. The Internationalization Process

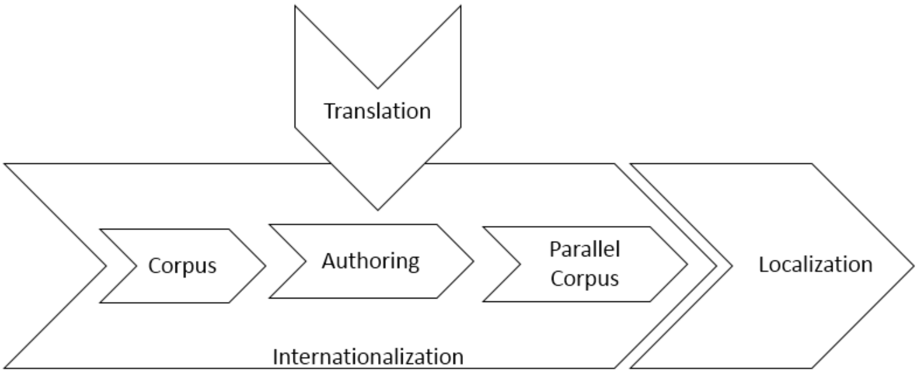
The internationalization process is a typical application of the *language industry* (LIND 2016), the technology sector dedicated to facilitating multilingual communication. In Section 2, the process of internationalization is rationalized and the technologies associated with its automation are discussed, before introducing the experimental implementation described in Section 3. The core requirements of an internationalization process are expressed by the following definition (Esselink and O'Brien, 2000): *Internationalization is the process of generalizing a product so that it can handle multiple languages and cultural conventions without the need to be re-designed. Internationalization takes place at the level of program design and document development*. Internationalization is achieved by providing an information system with the following features:

- A *metadata schema* that allows the language of the content to be specified, and a stylesheet that allows vertical or bidirectional orientations of texts, and incorporates specific formats for the representation of date and time.
- Content available in several languages. All possible translations need not be included when the document is created, but it is important to adopt a *standard* that allows for the later addition of other languages as needed.
- A *character encoding* suitable for all the languages handled by the system.

Localization is another concept related to internationalization. As stated by (Esselink and O'Brien, 2000): “*Localization means taking a product and making it linguistically and culturally appropriate to the target locale (country/region and language) where it will be used and sold”*. In the modern language industry, localization is always included in the design of an internationalization process (Schaler, 2010; Ishida and Miller, 2016). Internationalization creates a product which is prepared in advance for localization, previously translated and able to be automatically adapted to demand. As a process, internationalization is defined by the following four elements: *activity, input, output* and *user*:

1. The *activity* of internationalization is made up of interrelated actions which aim to produce digital documents that are accessible to multilanguage users, and which add value in terms of a cost-benefit ratio.
2. The *input* to the process is typically a collection of digital documents which cover different subjects or which are from a common domain (i.e. a Spatial Data Infrastructure repository), and which are *multilingual* but not yet translated. The technical literature refers to this type of collection as a *multilingual corpus* (Simoes, 2004; Fernandes, 2012).
3. The *output* is the *multilingual parallel aligned corpus* (hereafter abbreviated to *parallel corpus*), which is a collection of documents in different languages where each of them has a translation. In the simplest case, only two documents and two languages are involved; one of the documents is an exact translation of the other, and one of the languages is the *source*, while the other is the *target* language. The parallel corpus is also defined as *aligned* if it is possible to identify the correspondences between the translations at the level of individual phrases with an exact mapping (Simoes, 2004).
4. For the *users*, the benefits brought by the internationalization process are unanimously acknowledged by the organizations that have adopted it (LTC, 2009; Steinberger et al., 2014; LIND, 2016), since this process:
    - Avoids costly and improvised reengineering at a later stage.
    - Prevents or reduces delays in the distribution of products.
    - Prevents the use of different approaches in different regions, which affects interoperability.
    - Develops greater business potential and gives visibility to the project.

Figure 1: Internationalization Value Chain



We can represent the internationalization process as a value chain (Figure 1). Note that translation (which can be human- or machine-generated) is introduced as an external resource, whereas internationalization adds value to the localization activity. Internationalization requires technological effort and process rationalization which anticipates how the multilingual corpora will be used by an organization. The technology used in internationalization, and more generally in the language industry, is borrowed from *document automation*, defined as support from IT systems for the construction of electronic documents, and is used in industry to increase productivity and the availability of information.

2.2. Literature Review of Standard Data Schema and Internationalization Systems

An internationalization system is an ensemble of software, data, standard formats and procedures. These elements are all relevant when designing the system. The product of the internationalization process is the parallel corpus, which for (Steinberger et al., 2012) is characterized by the following features: *language coverage, translation quality, usage of document identifiers, subject domain categorization* and *alignment granularity*.

The technological advancement of internationalization systems has taken place through the progressive integration of new functions; *corpora alignment, embedded translations, localization*, and standard schemas have been created where necessary to support these functions. Automation technology, which reduces human intervention where complex and repetitive actions are required, has been the main driving force behind this progress. The internationalization and localization processes have gradually evolved from an approach based mainly on human work, for both document translation and formatting, to a predominantly automated process. These functions are also implemented in the system discussed in this paper, which automatically produces a parallel corpus. There are two technologies relevant to this project and which enable automation: the *metadata schema and metadata processors*.

Older systems produce parallel corpora from text files. A piece of text is broken down into phrases, translated and aligned either manually or based on statistical algorithms. Other systems, such as the one presented in this paper, process XML files instead, in which the text is accompanied by descriptive metadata. The XML schema (W3C recommendation) allows the machines to navigate automatically through documents. The metadata processor is the software that navigates the metadata and automatically and selectively extracts content on the basis of the tags, composing new documents where necessary. Schemas can be specialized to support specific features. In internationalization systems, schemas allow the addition of metadata and translations to the document to achieve parallelization, alignment and localization (Sasaki, 2009).

The *XSLT (Extensible Stylesheet Language Transformation)*[3] utilized in this work and *XQuery*[4] technologies are programming language platforms used to develop metadata processors. Both platforms use the *XPath* function, which allows automatic navigation of the XML schema and the identification of tags. However, it should be remembered that a metadata processor is a scripting program specialized in a specific schema; it is therefore necessary to adapt the program whenever an XML document deviates from that schema. The adaptation of the metadata processor requires consistent effort; this becomes substantial when the corpus is large, and is totally impractical for real-time applications and productive processes. Here, the adoption of a standard format for multilingual documents is used in automation. A standard format is a *non-functional requirement* of all systems contributing to an open information exchange network. Standards used in internationalization include an *XML schema* and a set of guidelines for designers who wish to create interoperable automatic systems (Fernandes, 2012; Wright, 2013).

Corpus Alignment

The simplest and most basic parallel corpus is a large collection of human-generated translations. An *aligner*, that is a metadata processor, processes these translations to turn them into a *parallel aligned corpus*. The parallel corpus is defined as *aligned* if it is possible to identify the correspondences between the translations at the level of individual phrases (Simoes, 2004). A longstanding standard format for aligned corpora is the *Text Encoding Initiative (TEI)*. This XML schema was originally defined to facilitate the exchange of electronic texts (Ahronheim, 1998). In its newer versions (Vanhoutte, 2004), TEI includes over 500 tags for textual analysis, some of which specifically support the construction of parallel aligned corpora. *JRC-Aquis* (Steinberger et al., 2006) is an example of a corpus based on the TEI standard, produced for the training and testing of text mining software. In JRC-Aquis, a corpus of multilingual but non-parallel documents are acquired from the web in HTML format. After the pre-processing needed to clean the data and transform it into XML documents, alignment is carried out. In accordance with the TEI standard, the final parallel corpus contains three separate files for each document and each pair of languages: the text in the source language, the text in the target language, and an alignment file which creates the parallel. JRC-Aquis uses two statistical aligners: *Vanilla* (Gale and Church, 1994) and *HunAligner* (Varga et al., 2007); these find aligned segments based on sentence length. The statistical aligners (Aswani, 2012) work by assigning a probabilistic score to each pair of sentences, according to the ratio of their lengths. At each iteration, the highest ranked pair is considered to be aligned. Due to their simplicity,

[3] https://www.w3.org/TR/xslt-30/
[4] https://www.w3.org/XML/Query/

statistical aligners generally represent the first choice when working on large numbers of texts, before more complex tools are attempted.

Embedded Translations

The *Translation Memory Exchange (TMX)* (LISA, 2003) is an XML schema that, unlike TEI, includes tags which allow the source and one or more translations to be *embedded* within the same document. A *Translation Memory* (TM) is a database based on the TMX standard, which stores text segments; these can be sentences or paragraphs (the granularity may vary) and their translations. Each source text and its corresponding translation constitute a language pair, called a *Translation Unit (TU)*. The TM constitutes an important step in the automation process; in practice, a TM allows the recycling of a previous translation in the course of new translation, resulting in a faster result and a higher quality product for the translation industry. Software programs that use translation memories are known as *Translation Memory Managers* (TMM). *DGT-TM* (Steinberger et al., 2012) is an example of a parallel aligned corpus based on TMX, which contains legislative documents from the body of European law. The DGT-TM contains tens of millions of language pairs in 24 languages. The sentence alignment is carried out using *Euramis* (Valli, 2012); this is a text search application on a network that includes a *concordancing tool*, the function of which is to find the best translation for a given string within a corpus. For alignment, Euramis makes use of anchors such as numbers and text numbering, as well as images and other non-linguistic information. Once alignment is complete, Euramis allows the user to look at the strings in the original documents as an additional quality control tool for the translation. The level of granularity of the DGT-TM corpus is usually full sentences, but can also be headings or full paragraphs. Using the TMM TMXtract, a TM is extracted from the DGT-TM corpus by specifying the source language and the target language to be used in translation jobs. DGT-TM has also been used to train the in-house *statistical machine translation* (SMT) MT@EC.

Support to Localization

In recent years, organizations have demonstrated a growing interest in sharing content in a global and interconnected world. Progress in internationalization technology has focused on the production of standards and systems that allow the continuous and automatic updating of parallelized texts, support for interoperability between systems using different standard schemas, and, above all, support for the automation of the *localization* process. The *XML Localization Interchange File Format (XLIFF)* (OASIS, 2007) is currently the most widespread standard for managing internationalization and localization simultaneously, and has been adopted by major commercial software companies, enterprise organizations and academia (Anastasiou, 2010). To support localization, XLIFF uses schemas containing additional information about the content (e.g. text directionality, coding, and other location-specific items) which were not considered in older standards. In order to support interoperability, in addition to the TMX schema properties, XLIFF provides an extensibility mechanism that allows the use of non-standard tags for compatibility with other schemas, enabling XLIFF to be is used as a container to import documents from other standard schemas. In this way, XLIFF allows an organization to construct an internationalization process using legacy software tools, with significant advantages in terms of cost containing and process automation. The *internationalization and localization system (ILS)* of (Pawar et al. 2015) is based on XLIFF, and also integrates an automatic translation system. The input is a source document that, once loaded, is converted into XLIFF format. The system is composed of two main components: *format extraction* and *format rebuilding*. Format extraction separates the contents of a document into two files; the first contains the part for translation, while the second contains the non-translatable part and the placeholders. Following translation, the format rebuilding phase reassembles the files (still in XLIFF format) by inserting additional tags for alignment and localization. Finally, ILS produces, on demand, localized versions of the document from XLIFF. Particularly interesting is the ability of ILS to automatically update embedded translations and localized versions when the source document is updated.

Further Evolution of Internationalization Technology

To complete this overview of internationalization technology, new coding schemas and systems for internationalization should be mentioned which are based on alternative technologies to XML, and which have been introduced into the world of internationalization in recent years in response to major IT innovations.

JavaScript Notation (JSON) is a format for data interchange on the web. In a similar way to XML, JSON supports the Unicode standard (The Unicode Consortium, 2006), which is indispensable for internationalization; however, the two standards are significantly different in other respects. JSON stores data in arrays, contains only text and numbers and is a compact format, while XML stores data in trees, can contain any type of data and is a verbose format. JSON is more suitable for applications where speed is required in terms of computation and data transmission; XML is recommended where document extensibility is required and integration with other markup languages such as HTML. The *European Data Portal*[5] *(EDP)* is an example of a system that uses the JSON format; this is an internationalized system that generates a parallelized corpus of metadata. EDP is similar in some respects to the MT@EC-Wrapper developed in this work. Each document harvested by the EDP is embedded in JSON format and sent to the European Commission's MT@EC machine translation service, which returns a

[5] https://www.europeandataportal.eu/

separate document for each target language that is integrated in the parallel corpus in a non-embedded format. An in-depth comparison of EDP and Wrapper is presented in Section 4.

In the *semantic web,* multilingual documents involve *multilingual linked data* (Gayo 2013): these are text segments distributed over the internet and semantically linked, so that they can be processed using a semantic query language such as *SPARQL,*[6] on which semantic data processors are based. The *Resource Description Framework* (*RDF)*[7] is the metalanguage of linked data; it contains multilingual labels that indicate the language of each text segment and the corresponding segments in different languages, so that a semantic search can be made which is independent of language. The RDF schema retains basic formatting information such as font sizes and styles, which are useful for localization. The DBpedia project (Auer et al., 2007) is an early example of a multilingual linked dataset. DBpedia aimed to convert the multilingual Wikipedia corpus to RDF and to publish it as Linked Open Data. In order to achieve parallelism, the *Inter-Language Links (ILL)* in Wikipedia are exploited. An ILL is a special link in a Wikipedia page that connects to the same or the closest possible page in terms of meaning, in a different language edition of Wikipedia. DBpedia's localization is still in progress, although the RDF schema already fully supports this feature (Gayo 2013).

2.3. Internationalization of Documents from the Geospatial Domain

The *International Organization for Standardization (ISO)* has defined several standards for annotating language resources (Laurent, 2015) that include the properties of the standards already mentioned in Section 2.2. The ISO standards also add domain-specialized schemas, such as that for geographic information, which is adopted in the Wrapper project and which is described in detail in this section. The *ISO TS 19103:2005* standard (ISO, 2015) specifies the *Unified Modelling Language (UML)* profile for modelling geographic information (Kresse and Fadaie, 2010). The standard provides guidelines on how UML should be used to create standardized geographic information and service models. The *ISO 19115* (ISO, 2014) standard of ISO/TC 211 defines a schema for describing geographic information and services by means of metadata, whereas the schemas discussed in previous sections are focused on general text. ISO 19115 contains information about the identification of digital geographic data and services, their extent, quality, spatial and temporal aspects, content, spatial reference, distribution and other properties. The *ISO 19139* (ISO, 2014) standard, adopted for the parallel corpus developed in this project, is an implementation of ISO 19115, and defines how metadata conforming to ISO 19115 should be represented in XML. The ISO also defines a set of namespaces in order to support the normalization of geographic

[6] https://www.w3.org/TR/sparql11-query/
[7] https://www.w3.org/RDF/

information (INSPIRE, 2009). ISO/TS 19139 is suitable for internationalization, and is recommended by INSPIRE as a schema for the interoperability of geographic information systems. One example of an internationalized system based on the ISO 19139 standard is Geonetwork;[8] this is already a component of the DRDSI infrastructure, and contains the corpora of this project. Geonetwork is an application catalogue for spatially-referenced data which has been adopted worldwide in Spatial Data Infrastructures. Geonetwork offers authoring functions and is able to manage and query internationalized documents using the ISO-19115 and ISO-19139 formats.

A detailed description follows about the transformation performed by the Wrapper's metadata processor, based on the ISO19139 standard. In a monolingual document, elements such the title and abstract contain an element called *CharacterString,* which in turn contains the text, as follows:

```
< title>
  < CharacterString> title of the document</CharacterString>
</title>
```

In a parallel aligned corpus adopting the embedded translation solution and following the ISO19139 standard, the multilingual element is *transformed* as follows (INSPIRE, 2009). Firstly, the target languages of the parallel elements are defined in the XML document (German is used in the following example):

```
<MD_Metadata>
  <locale>
    <PT_Locale id="DE">
      <languageCode>
        <LanguageCode codeList=http://www.loc.gov/standards/iso639-2/ codeListValue ="DE">
        </languageCode>
      </characterEncoding/>
    </PT_Locale>
  </locale>
....
```

These languages are defined so that they can later be used in the same document to indicate the corresponding translations:

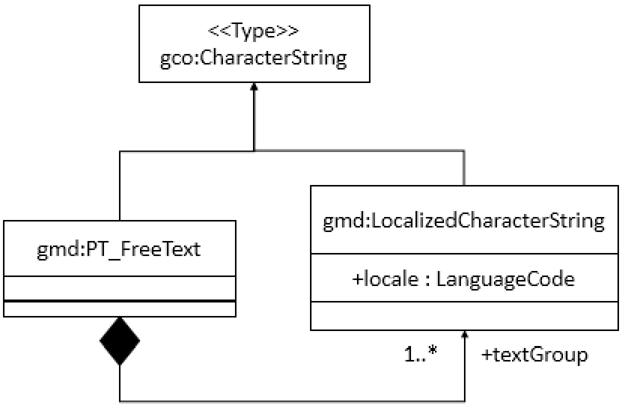
```
<title type="PT_FreeText_PropertyType”>
  <CharacterString> title of the document </CharacterString>
    <PT_FreeText>
      <textGroup>
        <LocalisedCharacterString locale="#IT">
             Titolo del documento
        </LocalisedCharacterString>
      </textGroup>
```

[8] http://geonetwork-opensource.org/

```
      <textGroup>
        <LocalisedCharacterString locale="#DE">
            Title des Dokuments
        </LocalisedCharacterString>
      </textGroup>
    </PT_FreeText>
  </title>
```

The *type* attribute indicates that the element title is not instantiated through a simple CharacterString, but rather as free text. Thus, the element title contains the sub-element *PT_FreeText,* which in turn contains one or more *textGroup* elements (one for each target language). Finally, each *LocalisedCharacterString* sub-element contains a translation of the string and attributes specifying the language, the country and the character encoding. The ISO 639-1 standard is used to indicate the language with a two-letter code (Sasaki, 2009).

Figure 2: UML Model for Geographic Information (ISO TS 19103:2005)



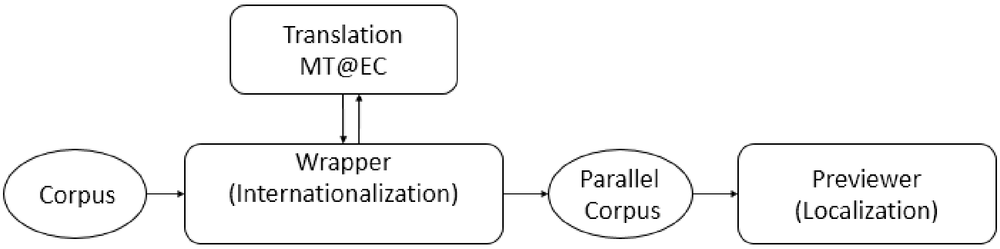
Source ISO, 2015

In Figure 2, the ISO 19139 schema adopted in the embedded translation is shown in a UML static diagram. The components of the schema and their hierarchy are represented as a set of classes and their relationships, according to the conceptual schema of ISO/TS 19103, using the namespaces defined by the standard.

3. INTERNATIONALIZATION OF THE DRDSI REPOSITORY

This chapter describes the architecture of the MT@EC-Wrapper (hereafter abbreviated to Wrapper). The Wrapper implements the internationalization process described in Section 2.1; its specific goal is to fully automate this process using a robust architectural solution. The process implemented by the Wrapper is shown in Figure 3; a multilingual corpus consisting of ISO19139 documents extracted from the DRDSI repository constitutes the *input*. This corpus is transformed by the Wrapper into a parallel corpus, the *output*. In the *process activity*, the wrapper produces a parallelized output with all embedded translations in a single document. In addition, the product document contains information for localization. These properties already exist in the TMX and XLIFF formats; however, it was decided that ISO19139 would be maintained as the output format in this project, since it includes all of these properties, is specialized in geographic metadata and is the format in which DRDSI already stores the metadata. The multilingual structure for the parallel corpus is that described in Section 2.3 for ISO19139. In carrying out the translations, the Wrapper interfaces automatically with the external translation service of MT@EC. The Wrapper also implements functions for the *Quality Control (QC)* of the product and the performance of the process, and uses a previewer module, which retrieves a document from the parallel corpus at the user's request and localizes it by building HTML pages in the required language.

Figure 3: Internationalization Process of the MT@EC-Wrapper



3.1. Project Requisites

The DRDSI system regularly collects metadata (*harvesting*) from remote catalogues available through *CSW,* and stores it locally. A total of 5,230 datasets are available within the DRDSI platform as of 12/08/2016. The collected metadata contain basic information about the dataset such as the *title, abstract, keywords* and *source organization*. The content of the element can be in any of the eight official languages of the project (plus English), and is not translated.

A preliminary requirement analysis was performed for this project. The DRDSI administrator requires a prototype software that entirely automates the process of constructing a parallel corpus, starting from the documents in their current status. In particular, for each element to be translated, the Wrapper has to extract it automatically from the XML, forwarding a translation request to the *MT@EC*[9] *on-*

[9] http://ec.europa.eu/dgs/translation/translationresources/machine_translation/index_en.htm

Line service of the European Commission, receiving the translation and integrating it into the parallel aligned corpus. In addition, the Wrapper must automatically verify the compliance of the source document and its internationalized version with the ISO19139 standard, and produce localized documents on request. The quality of the translation is that currently offered by the MT@EC service, and improvements to this are outside the scope of the project at this stage. The design of the Wrapper was also subject to non-functional requirements for integration into the existing DRDSI system. This required:

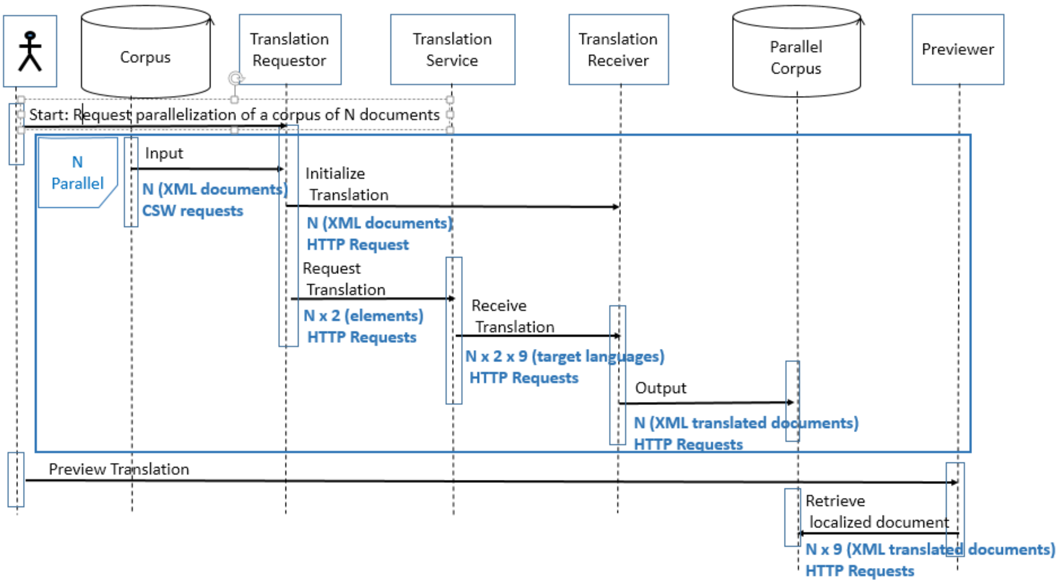
- The selection of the ISO19139 standard, already adopted by the DRDSI system as the format for the acquisition of metadata.
- The adoption of a service-oriented architecture based on HTTP, for distributing the Wrapper over a local network, which is also segmented into different subnets.

It should be pointed out that in its current configuration, the Wrapper produces corpora in which the parallelization is limited to the title and the abstract elements of the XML document, and that alignment is at the level of XML elements. This was sufficient to demonstrate the automation of all operations. The localization produced by the previewer is also limited to internal use. An extension of the system towards finer granularity in the alignment and parallelism extending to other elements of the ISO1915 schema, such as keywords, would require a minor system reconfiguration without involving substantial changes to the architecture.

3.2. Processing Sequence

The simplified *sequence diagram* in Figure 4, taken from the UML documentation of the project, shows the order of events in the interactions between the software objects in the main use case: a request for the parallelization of a document. The vertical dashed lines represent the life line of an object. Within its period of activity, each object generates messages by which it communicates with other objects. The diagram also shows the range of messages exchanged when the user requests the parallelization of *N* documents.

Figure 4: Wrapper Sequence Diagram of Parallelization


Corpus
Translation Requestor
Translation Service
Translation Receiver
Parallel Corpus
Previewer
Start: Request parallelization of a corpus of N documents
N Parallel
Input
N (XML documents) CSW requests
Initialize Translation
N (XML documents) HTTP Request
Request Translation
N x 2 (elements) HTTP Requests
Receive Translation
N x 2 x 9 (target languages) HTTP Requests
Output
N (XML translated documents) HTTP Requests
Preview Translation
Retrieve localized document
N x 9 (XML translated documents) HTTP Requests

The sequence, from the request to the creation of a parallel corpus, is detailed below:

1. *Start*: The user starts the process by requesting the parallelization of a corpus of *N* documents.
2. *Input*: The input to the process is a corpus of ISO-19139 documents taken from the DRDSI repository. Each document is retrieved with an individual *CSW* query. *N* independent processes of parallelization are initialized.
3. *Initialize translation*: The *requestor object* sends a copy of the monolingual document to the receiver.
4. *Request translation*: The requestor module extracts the contents of the *title* and *abstract* from the corresponding XML elements. The content of each distinct element is sent separately to the translation service, via an HTTP request. A total of 2*N* requests are sent to MT@EC. Each request indicates the target language of the translation; the same message can request translations in several target languages (up to the nine languages of the project).
5. *Receive translation*: The MT@EC translation service produces a separate response message for each item of translated content and for each target language (totalling 2*N* x 9 messages), and transmits them to the receiver.
6. *Output*: The receiver extracts the translated content from each response message. After identifying the content, the receiver writes it in a local copy of the document, progressively creating a parallel aligned corpus of *N* new documents. The parallel corpus is saved in the Wrapper's local DBMS (which

is distinct from the DRDSI repository) during the process, until after completion, when the user decides whether to move it into the DRDSI repository. The parallelized document follows the ISO-19139 standard for multilingualism.

Additional available functions are *document QC* and *process performance monitoring*. Figure 4 shows the sequence of translation previewing, as follows:

7. *Translation preview*: the user requests the previewer to view a document in a given language, chosen from the nine available.
8. *Retrieve localized document*: the previewer retrieves the document from the parallel corpora; it extracts the content in the specified language only.

The Wrapper was tested on the entire repository. A corpus of 5,230 multilingual documents, with total size of about 100GB, was parallelized. Several exceptions were implemented in the code to manage the inconsistencies frequently found in the schema of the documents, thus increasing the robustness. The success rate of the document processing was 98%.

3.3. A Microservices Architecture to Handle Process Complexity

A basic requirement of the Wrapper is its modularity, which is necessary for software extensibility, to provide diversification of services and utilities, and to deploy the software on a highly segmented local network. This leads to the implementation of a RESTful *Service Oriented Architecture (SOA)*; this consists of services accessible through the HTTP protocol, the most firewall-friendly protocol. Compared to alternative agent-based service technologies, REST does not require intermediary brokerage services. A particular type of SOA was adopted for the Wrapper: the microservices architecture (Wolff, 2016), consisting of the implementation of every single function in a distinct network-exposed service. This architecture provides several further advantages:

- *Processing parallelism*: Microservices architecture exploits web server capabilities to contain multiple competing service requests and processes. This means the Wrapper need not implement job queue functions. During the process, a set of competing service requests are instantiated for multiple text elements, and are executed in parallel. In addition, microservices manage the submission of multiple competing requests to the external MT@EC service. *Software robustness*: Microservices allow the isolation of errors and exceptions in the complex sequence of operations of each process, without interfering with other processes in progress.
- Microservices software modules can be distributed as necessary to particular nodes in the network.

- *Agile development*: Distributing the responsibilities of the system among smaller services made easier, during the development work, to add functions and to improve fault isolation and robustness.

Figure 5: Wrapper Architecture and Deployment

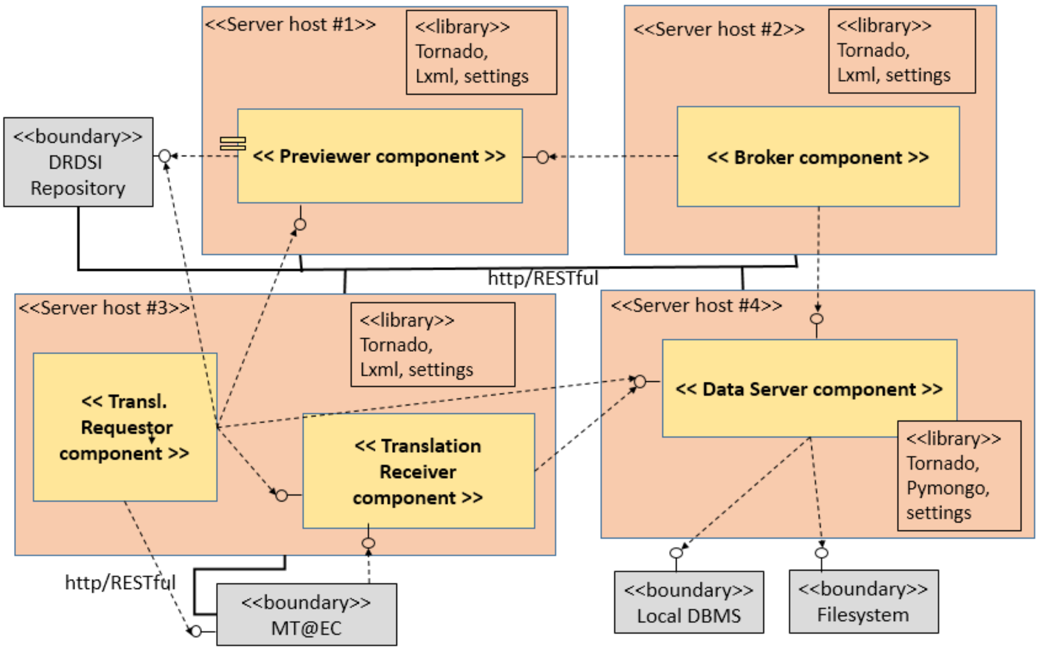

<<Server host #1>>
<<library>> Tornado, Lxml, settings
<< Previewer component >>
<<Server host #2>>
<<library>> Tornado, Lxml, settings
<< Broker component >>
<<boundary>> DRDSI Repository
http/RESTful
<<Server host #3>>
<<library>> Tornado, Lxml, settings
<< Transl. Requestor component >>
<< Translation Receiver component >>
<<Server host #4>>
<< Data Server component >>
<<library>> Tornado, Pymongo, settings
http/RESTful
<<boundary>> MT@EC
<<boundary>> Local DBMS
<<boundary>> Filesystem

Figure 5 is taken from the UML documentation of the project, and shows the complete architecture of the Wrapper system, using a simplified *deployment diagram*. The diagram also shows the dependencies between packages, the communication mode, the external components (with the *<<boundary>>* stereotype) and the host node where each component is located. It should be noted that no component of the Wrapper acts as a controller for the other components. A component has a graphical user interface (GUI) if it has functions that require user intervention; for example, the Translation Requestor component has a GUI for requesting internationalization, while the Broker has a GUI for process monitoring. At function level, the components are totally independent. Each user's request triggers a sequence of service requests between the components; a component carries out its services or invokes services from other components, and each request is asynchronous. The communication with external services and between components is RESTful and is built on *Tornado*, which is a web framework and asynchronous networking Python library. In particular, asynchrony is needed to manage possible delays in the *MT@EC response* and to prevent cases where the failure of a single translation compromises the execution of an entire job with multiple documents. The *HTTP Post* method was used to implement communication between microservices. The following paragraphs detail the technical characteristics of the main components and their interactions.

DRDSI Repository

The DRDSI repository is the external component containing the non-parallel multilingual corpus. The Translation Requestor component extracts the documents from the repository, via the CSW service, and sends them to the internationalization procedure. The Previewer component also accesses the DRDSI repository to display the documents at the user's request. *Geonetwork*, an instance of which implements the DRDSI metadata repository, requires that the *PT_FreeText* and *LocalisedCharacterString* elements described in Section 2.3 are used in the metadata schema for multilingual content indexing.

MT@EC Service

The Translation Requestor component invokes the MT@EC translation service, which returns the completed translations to the Translation Receiver component. *MT@EC* is an external resource that is freely available to European Commission staff as well as staff from the public administrations of EU Member States. MT@EC is a *translation machine*, which can translate formatted and plain text documents from any one official EU language into another. MT@EC makes available a URL, ready to accept a message transmitted with the HTTP POST method, containing the string of text to be translated. MT@EC responds only to *asynchronous* non-blocking service requests; this is to avoid the situation where communicating parties remain blocked while waiting for answers from each other. After processing the request, MT@EC responds with a message containing the translated string. In its communications, the MT@EC service uses the REST protocol for packaging the requests, and XML for messages. The interchangeability between MT@EC and other similar services depends on these formats. The Google Translate API[10]() also uses the same formats, and could therefore easily replace MT@EC in the Wrapper. Other translation services exist which use SOAP or JavaScript for packaging requests and the JSON format for messages, and therefore cannot be used by the Wrapper unless specific Receiver and Translation modules enabled for those formats are implemented.

Translation Requestor

The Translation Requestor component implements the translation request function. In the internationalization process, it communicates with the following components:

[10] https://cloud.google.com/translate/

- The DRDSI repository, from which it retrieves the document to be internationalized;
- MT@EC, to which it submits the translation request; and
- The data server, which initializes a new internationalized document in the Wrapper's local DBMS.

Using the GUI of this component, the user can request translations for a single document, for multiple documents at the same time (batch processing mode), or for the entire repository at once. The Requestor implements an *XML parser*, which uses the *LXML* Python Library. The LXML library is based on the *libxml2* and *libxslt* C libraries. In the technical literature, LXML is considered to be among the best performing solutions for metadata processing (Sourceforge, 2009). The parser extracts the content to be translated from the document; more specifically, the *parse()* method of LXML returns the full document retrieved from GeoNetwork by calling its CSW service, while *XPath()* is a XML tree navigation and element content extraction method. Each piece of extracted content is sent to the translation system using an HTTP request.

Translation Receiver

This component implements the sub-process of receiving the translation. In the internationalization process, this component communicates with:

- MT@EC, from which it receives the translations; and
- The data server, to save and update the document being transformed in the Wrapper's local DBMS.

The Translation Receiver listens on a host port, waiting for *response messages* from the MT@EC server. A handler is activated to manage HTTP POST requests from MT@EC. In the receiver, the LXML library is used to implement the metadata processor, to update the parallel corpus by writing the string of each received translation into the position assigned by the ISO-19139 standard, and to add extra elements to the XML document tree when needed. Internationalization requires the adoption of character encoding to represent the linguistic symbols of all languages. The objective of the Unicode standard (The Unicode Consortium, 2006) is to cover the characters of all the alphabets of the world, independently of the platform and software used. The Unicode standard also provides various encoding forms that allow the use of more compact codes for the most frequently used characters. *Eight-bit Unicode Transformation Format (UTF-8)* encoding was chosen for the Wrapper, as this can represent characters with sequences of a single byte, leading to smaller memory requirements.

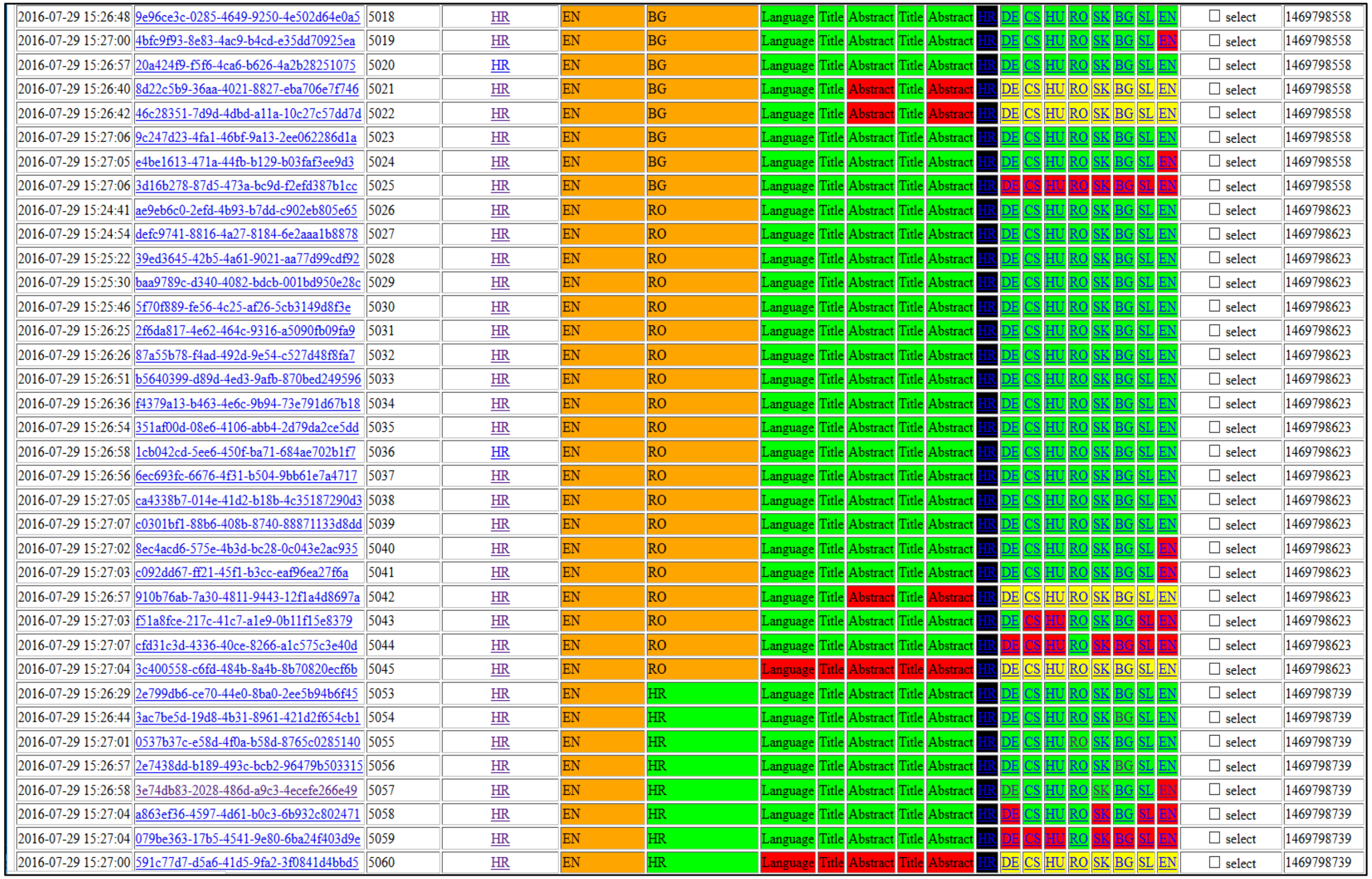
Data Server

This component implements methods for accessing the parallel corpus temporarily stored in the Wrapper's local DBMS. The data server receives requests via RESTful calls from other components to perform queries on the documents. The document-oriented database MongoDB was chosen as the technology for the local DBMS. MongoDB is a *NoSQL* solution that relies on the internal structure of the stored document in order to extract data. A NoSQL document contains a parallelized multilingual document along with information indicating its processing state. The choice of NoSQL was motivated by the need for a high-performance system with backup/recovery functions, and by the fact that a schemaless database is sufficient for the intended functionalities of the wrapper. In addition, MongoDB is directly integrated in Python through the PyMongo library.

Broker

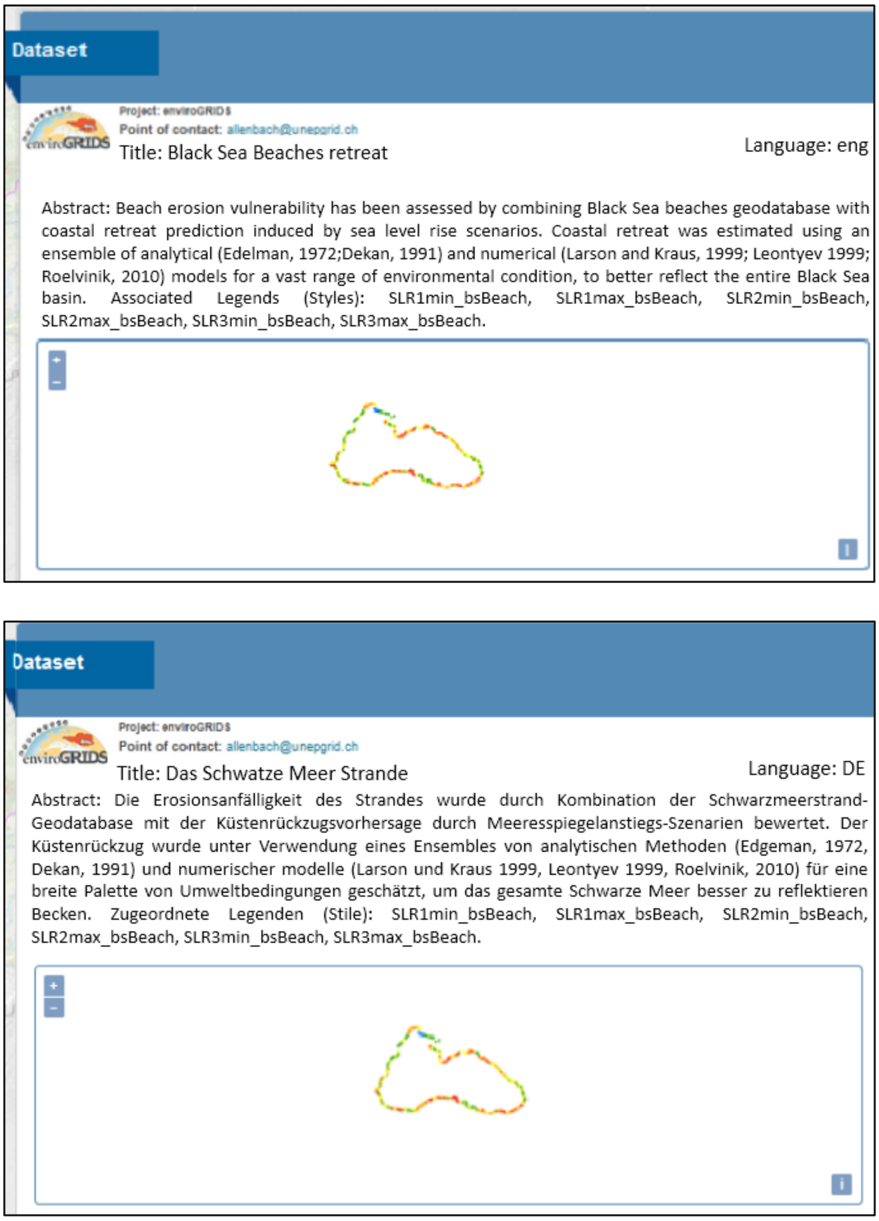
The Broker component implements the process monitoring function. The Broker retrieves and displays information about the status of each document in process, using a tabular GUI (Figure 6) which is continuously updated. The Broker does not have connections to the Translation Requestor and Receiver modules; it communicates with the data server, through which it periodically reads information about the status of the document. The broker also communicates with the previewer when the user requests viewing of the product document. In Figure 6, each row of the GUI corresponds to a document parallelization process. From left to right, the fields contain the date-time of the last update, a link to a preview of the original document, the document identifier, the source language, the actual language of the title and abstract elements, QC information, the status of notification from MT@EC, links to separate localized previews of the multilingual corpus in the nine official languages of the project, commands for deleting and exporting documents, and the process identifier. The colours of the QC fields indicate one of three possible statuses: successful completion (*green*), incomplete (*yellow*), or unsuccessful (*red*).

Figure 6: Broker Graphic User Interface



2016-07-29 15:26:48	9e96ce3c-0285-4649-9250-4e502d64e0a5	5018	HR	EN	BG	Language	Title	Abstract	Title	Abstract	HR	DE	CS	HU	RO	SK	BG	SL	EN	□ select	1469798558
2016-07-29 15:27:00	4bfc9f93-8e83-4ac9-b4cd-e35dd70925ea	5019	HR	EN	BG	Language	Title	Abstract	Title	Abstract	HR	DE	CS	HU	RO	SK	BG	SL	EN	□ select	1469798558
2016-07-29 15:26:57	20a424f9-f5f6-4ca6-b626-4a2b28251075	5020	HR	EN	BG	Language	Title	Abstract	Title	Abstract	HR	DE	CS	HU	RO	SK	BG	SL	EN	□ select	1469798558
2016-07-29 15:26:40	8d22c5b9-36aa-4021-8827-eba706e7f746	5021	HR	EN	BG	Language	Title	Abstract	Title	Abstract	HR	DE	CS	HU	RO	SK	BG	SL	EN	□ select	1469798558
2016-07-29 15:26:42	46c28351-7d9d-4dbd-a11a-10c27c57dd7d	5022	HR	EN	BG	Language	Title	Abstract	Title	Abstract	HR	DE	CS	HU	RO	SK	BG	SL	EN	□ select	1469798558
2016-07-29 15:27:06	9c247d23-4fa1-46bf-9a13-2ee062286d1a	5023	HR	EN	BG	Language	Title	Abstract	Title	Abstract	HR	DE	CS	HU	RO	SK	BG	SL	EN	□ select	1469798558
2016-07-29 15:27:05	e4be1613-471a-44fb-b129-b03faf3ee9d3	5024	HR	EN	BG	Language	Title	Abstract	Title	Abstract	HR	DE	CS	HU	RO	SK	BG	SL	EN	□ select	1469798558
2016-07-29 15:27:06	3d16b278-87d5-473a-bc9d-f2efd387b1cc	5025	HR	EN	BG	Language	Title	Abstract	Title	Abstract	HR	DE	CS	HU	RO	SK	BG	SL	EN	□ select	1469798558
2016-07-29 15:24:41	ae9eb6c0-2efd-4b93-b7dd-c902eb805e65	5026	HR	EN	RO	Language	Title	Abstract	Title	Abstract	HR	DE	CS	HU	RO	SK	BG	SL	EN	□ select	1469798623
2016-07-29 15:24:54	defc9741-8816-4a27-8184-6e2aaa1b8878	5027	HR	EN	RO	Language	Title	Abstract	Title	Abstract	HR	DE	CS	HU	RO	SK	BG	SL	EN	□ select	1469798623
2016-07-29 15:25:22	39ed3645-42b5-4a61-9021-aa77d99cdf92	5028	HR	EN	RO	Language	Title	Abstract	Title	Abstract	HR	DE	CS	HU	RO	SK	BG	SL	EN	□ select	1469798623
2016-07-29 15:25:30	baa9789c-d340-4082-bdcb-001bd950e28c	5029	HR	EN	RO	Language	Title	Abstract	Title	Abstract	HR	DE	CS	HU	RO	SK	BG	SL	EN	□ select	1469798623
2016-07-29 15:25:46	5f70f889-fe56-4c25-af26-5cb3149d8f3e	5030	HR	EN	RO	Language	Title	Abstract	Title	Abstract	HR	DE	CS	HU	RO	SK	BG	SL	EN	□ select	1469798623
2016-07-29 15:26:25	2f6da817-4e62-464c-9316-a5090fb09fa9	5031	HR	EN	RO	Language	Title	Abstract	Title	Abstract	HR	DE	CS	HU	RO	SK	BG	SL	EN	□ select	1469798623
2016-07-29 15:26:26	87a55b78-f4ad-492d-9e54-c527d48f8fa7	5032	HR	EN	RO	Language	Title	Abstract	Title	Abstract	HR	DE	CS	HU	RO	SK	BG	SL	EN	□ select	1469798623
2016-07-29 15:26:51	b5640399-d89d-4ed3-9afb-870bed249596	5033	HR	EN	RO	Language	Title	Abstract	Title	Abstract	HR	DE	CS	HU	RO	SK	BG	SL	EN	□ select	1469798623
2016-07-29 15:26:36	f4379a13-b463-4e6c-9b94-73e791d67b18	5034	HR	EN	RO	Language	Title	Abstract	Title	Abstract	HR	DE	CS	HU	RO	SK	BG	SL	EN	□ select	1469798623
2016-07-29 15:26:54	351af00d-08e6-4106-abb4-2d79da2ce5dd	5035	HR	EN	RO	Language	Title	Abstract	Title	Abstract	HR	DE	CS	HU	RO	SK	BG	SL	EN	□ select	1469798623
2016-07-29 15:26:58	1cb042cd-5ee6-450f-ba71-684ae702b1f7	5036	HR	EN	RO	Language	Title	Abstract	Title	Abstract	HR	DE	CS	HU	RO	SK	BG	SL	EN	□ select	1469798623
2016-07-29 15:26:56	6ec693fc-6676-4f31-b504-9bb61e7a4717	5037	HR	EN	RO	Language	Title	Abstract	Title	Abstract	HR	DE	CS	HU	RO	SK	BG	SL	EN	□ select	1469798623
2016-07-29 15:27:05	ca4338b7-014e-41d2-b18b-4c35187290d3	5038	HR	EN	RO	Language	Title	Abstract	Title	Abstract	HR	DE	CS	HU	RO	SK	BG	SL	EN	□ select	1469798623
2016-07-29 15:27:07	c0301bf1-88b6-408b-8740-88871133d8dd	5039	HR	EN	RO	Language	Title	Abstract	Title	Abstract	HR	DE	CS	HU	RO	SK	BG	SL	EN	□ select	1469798623
2016-07-29 15:27:02	8ec4acd6-575e-4b3d-bc28-0c043e2ac935	5040	HR	EN	RO	Language	Title	Abstract	Title	Abstract	HR	DE	CS	HU	RO	SK	BG	SL	EN	□ select	1469798623
2016-07-29 15:27:03	c092dd67-ff21-45f1-b3cc-eaf96ea27f6a	5041	HR	EN	RO	Language	Title	Abstract	Title	Abstract	HR	DE	CS	HU	RO	SK	BG	SL	EN	□ select	1469798623
2016-07-29 15:26:57	910b76ab-7a30-4811-9443-12f1a4d8697a	5042	HR	EN	RO	Language	Title	Abstract	Title	Abstract	HR	DE	CS	HU	RO	SK	BG	SL	EN	□ select	1469798623
2016-07-29 15:27:03	f51a8fce-217c-41c7-a1e9-0b11f15e8379	5043	HR	EN	RO	Language	Title	Abstract	Title	Abstract	HR	DE	CS	HU	RO	SK	BG	SL	EN	□ select	1469798623
2016-07-29 15:27:07	cfd31c3d-4336-40ce-8266-a1c575c3e40d	5044	HR	EN	RO	Language	Title	Abstract	Title	Abstract	HR	DE	CS	HU	RO	SK	BG	SL	EN	□ select	1469798623
2016-07-29 15:27:04	3c400558-c6fd-484b-8a4b-8b70820ecf6b	5045	HR	EN	RO	Language	Title	Abstract	Title	Abstract	HR	DE	CS	HU	RO	SK	BG	SL	EN	□ select	1469798623
2016-07-29 15:26:29	2e799db6-ce70-44e0-8ba0-2ee5b94b6f45	5053	HR	EN	HR	Language	Title	Abstract	Title	Abstract	HR	DE	CS	HU	RO	SK	BG	SL	EN	□ select	1469798739
2016-07-29 15:26:44	3ac7be5d-19d8-4b31-8961-421d2f654cb1	5054	HR	EN	HR	Language	Title	Abstract	Title	Abstract	HR	DE	CS	HU	RO	SK	BG	SL	EN	□ select	1469798739
2016-07-29 15:27:01	0537b37c-e58d-4f0a-b58d-8765c0285140	5055	HR	EN	HR	Language	Title	Abstract	Title	Abstract	HR	DE	CS	HU	RO	SK	BG	SL	EN	□ select	1469798739
2016-07-29 15:26:57	2e7438dd-b189-493c-bcb2-96479b503315	5056	HR	EN	HR	Language	Title	Abstract	Title	Abstract	HR	DE	CS	HU	RO	SK	BG	SL	EN	□ select	1469798739
2016-07-29 15:26:58	3e74db83-2028-486d-a9c3-4ecefe266e49	5057	HR	EN	HR	Language	Title	Abstract	Title	Abstract	HR	DE	CS	HU	RO	SK	BG	SL	EN	□ select	1469798739
2016-07-29 15:27:04	a863ef36-4597-4d61-b0c3-6b932c802471	5058	HR	EN	HR	Language	Title	Abstract	Title	Abstract	HR	DE	CS	HU	RO	SK	BG	SL	EN	□ select	1469798739
2016-07-29 15:27:04	079be363-17b5-4541-9e80-6ba24f403d9e	5059	HR	EN	HR	Language	Title	Abstract	Title	Abstract	HR	DE	CS	HU	RO	SK	BG	SL	EN	□ select	1469798739
2016-07-29 15:27:00	591c77d7-d5a6-41d5-9fa2-3f0841d4bbd5	5060	HR	EN	HR	Language	Title	Abstract	Title	Abstract	HR	DE	CS	HU	RO	SK	BG	SL	EN	□ select	1469798739

Figure 7: Local Versions of the Parallelized Document



Project: enviroGRIDS
Point of contact: allenbach@unepgrid.ch
Title: Black Sea Beaches retreat

Language: eng

Abstract: Beach erosion vulnerability has been assessed by combining Black Sea beaches geodatabase with coastal retreat prediction induced by sea level rise scenarios. Coastal retreat was estimated using an ensemble of analytical (Edelman, 1972;Dekan, 1991) and numerical (Larson and Kraus, 1999; Leontyev 1999; Roelvinik, 2010) models for a vast range of environmental condition, to better reflect the entire Black Sea basin. Associated Legends (Styles): SLR1min_bsBeach, SLR1max_bsBeach, SLR2min_bsBeach, SLR2max_bsBeach, SLR3min_bsBeach, SLR3max_bsBeach.

Project: enviroGRIDS
Point of contact: allenbach@unepgrid.ch
Title: Das Schwatze Meer Strande

Language: DE

Abstract: Die Erosionsanfälligkeit des Strandes wurde durch Kombination der Schwarzmeerstrand-Geodatabase mit der Küstenrückzugsvorhersage durch Meeresspiegelanstiegs-Szenarien bewertet. Der Küstenrückzug wurde unter Verwendung eines Ensembles von analytischen Methoden (Edgeman, 1972, Dekan, 1991) und numerischer modelle (Larson und Kraus 1999, Leontyev 1999, Roelvinik, 2010) für eine breite Palette von Umweltbedingungen geschätzt, um das gesamte Schwarze Meer besser zu reflektieren Becken. Zugeordnete Legenden (Stile): SLR1min_bsBeach, SLR1max_bsBeach, SLR2min_bsBeach, SLR2max_bsBeach, SLR3min_bsBeach, SLR3max_bsBeach.

Previewer

The Previewer component creates an HTML version of a document. Preview requests can come from:

- The *Translation Requestor*, when the user requests the visualization of the original documents stored in the DRDSI Repository, using the GUI of the Translation Requestor, before starting internationalization.
- The *Broker*, when the user requests the visualization of the internationalized documents stored in the Wrapper's local database, using the GUI of the Translation Requestor, in a specified language.

To accomplish these tasks, the Previewer component implements a complex *XSLT* parameterized timesheet, described above in Section 2.2, containing the instructions by which the *etree.XSLT* function of the LXML library transforms the retrieved parallel document into a localized HTML page. The previewer also can access external repositories on the Internet, provided they expose a CSW interface. Figure 7 shows two localized versions built from a parallel aligned corpus.

3.4. Distributing the Application

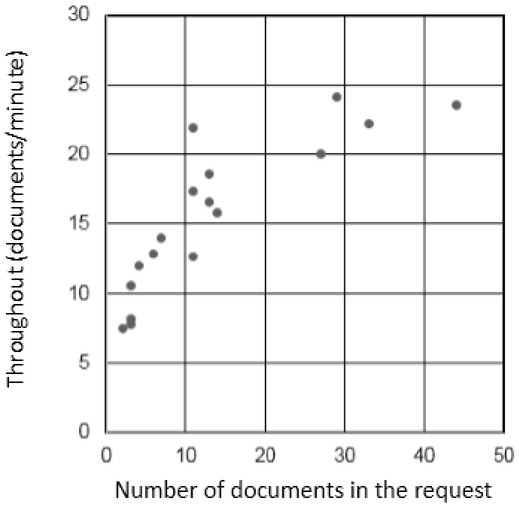
The application is distributed over different nodes. The service-oriented architecture also fulfils the need to distribute the functions appropriately among the security zones of the network infrastructure. In particular, the Receiver module, whose services are invoked from MT@EC, must be located in a security zone where it is visible from the Internet; all other modules can be distributed according to functional needs. The distribution also meets application scalability requirements, which allow the Wrapper to be reconfigured for use within other projects with multiple and personalized broker components, a Previewer, and interfaces for translation requests, which are variously distributed and which can communicate.

3.5. Level of Performance

As a non-functional requirement of the project, the Wrapper must have a level of performance that is compatible with the performance of the external MT@EC with which it interfaces. In simpler words, the Wrapper must not send translation requests at a rate that exceeds the response capacity that the MT@EC service. Thanks to the parallelism of its computational resources, the MT@EC system can receive multiple competing translation requests from a client and process them simultaneously. However the number of parallel translations performed for a given client have maximum limit that depends primarily on the quotas and the priority that the MT@EC administration assigns to the specific client, and also depends on the average length of the texts to translate, on the source and target languages, and the current overall workload of MT@EC system. On the other hand, to fully exploit the parallelism offered MT@EC, the Wrapper client must request multiple translations simultaneously and at the maximum rate permitted. A capacity planning study was necessary in order to evaluate experimentally the limit of parallel requests that MT@EC accepts and consequently to fix the Wrapper's request rate. In order to facilitate and simplify the performance analysis, a simplified operational model was applied consisting of a single service centre at the MT@EC, and a generator of translation requests, the Wrapper. The performance of the MT@EC service was evaluated in a series of load tests, conducted under the simplifying assumption that the performance of the MT@EC service was that exposed to the Wrapper.

In the first series of load tests, it was observed that the Wrapper's response time increased exponentially with the number of characters in the text to be translated. The *service times* were measured as follows: for text up to 1,000 characters, the MT@EC service times ranged between 10 and 200 seconds. However, the *response times* for the operations of loading the document from the DRDSI repository and extracting the string from the document were 0.3 s and 2 s on average, respectively. Due to its longer response time, the MT@EC resource represents the computational bottleneck of the system. The upper limit to the workload of the MT@EC service was defined experimentally by measuring the *throughput* (the number of service requests served in unit time) of this resource for an increasing number of concurrent requests (Figure 8).

Figure 8: MT@EC-Wrapper Process Throughput



The load limit, beyond which the throughput stops increasing, appears experimentally to be 20 documents per minute (or one document every three seconds). This limit value is considered to be the optimal rate for translation requests, beyond which the request queue at MC@EC grows indefinitely, causing system instability. Moreover, when submitting translation requests concurrently to MT@EC, the average residence time of a request apparently increases with the overall number of requests. From the point of view of the reliability of the service offered by the Wrapper, a situation whereby translations remain pending for a long time is undesirable, since this increases the risk that some translations would remain incomplete if any component of the system, or the MT@EC service itself, were to fail. The limit placed on the request rate also helps to reduce this risk.

In general, the tests show that the key parameter controlling the overall performance of the system is the *translation request submission rate*. By default, a rate value was set cautiously at 5 s (i.e. a request sent every 5 s), which allows the completion of 100 translations in less than 10 minutes. The user can change this value.

4. COMPARISON OF SYSTEMS AND DISCUSSION

In Table 1, several systems reported in the literature are compared with the Wrapper. The columns represent the features of the systems, and the systems are sorted by year of publication to highlight the trends. An analysis is worthwhile of certain key differences, as follows:

- *Input formats*: The JRC-Aquis, DGT-TM, and DBpedia systems can receive input documents with diverse formats, acquired from the web, or in a specific text format. This requires labour-intensive activity or using specific pre-processing tools in order to extract the texts to be internationalized. In contrast, the more recent systems of ILS, EDP, and Wrapper receive input which is already structured into metadata schemas. This is the consequence of the general increase in interoperability among modern information systems.
- *Granularity Alignment*: JRC-Aquis, DGT-TM and ILS have a granularity at sentence level, and use statistical aligners. For the other systems, the degree of granularity extends to the maximum capacity provided by the metadata schema for distinguishing text elements. Nominally, the Wrapper can align the contents of each element of the ISO schema.
- *Internationalization (Parallelization) Schema*: In the cases mentioned, each system uses a distinct schema for the parallel corpus. Newer systems such as the Wrapper, adopt schemas that support localization. In the most innovative systems, such as DBpedia, the corpus is a linked multilingual dataset, shared over the web. To this end, the Wrapper is integrated with the DRDSI, which converts the multilingual parallel corpus from ISO to RDF and publishes it as Linked Open Data.

Table 1: Comparison of Features of Internationalization Systems

	Input Document Format	**Alignment Granularity**	**Intern. Parallel. Schema**	**Embedded Translation**	**Localization**
JRC-Aquis (2006)	Text Files	sentence	TEI		
DGT-TM (2007)	Web pages	sentence	TMX	■	
DBpedia (2008)	Web pages	document	RDF		■
EDP (2015)	XML	document	XML		■
ILS (2015)	XML	sentence	XLIFF	■	■
MC@EC-Wrapper (2016)	XML	XML element	XML ISO19139	■	■

A detailed comparison of the Wrapper is possible with a similar system, the *European Data Portal (EDP),* described in Section 2.2; the Wrapper differs from this system in certain important respects. EDP sends an integral document to MT@EC for translation, and produces as many distinct copies of the source document as the number of target languages; the alignment is only at the level of the document. In contrast, the Wrapper decomposes the source document into text segments, which are sent separately for translation, and then recomposes them. When the process is complete for each source document, the wrapper produces a parallel aligned document with all translated segments embedded in a single file. A further difference is that EDP uses the FTP protocol to send documents to the MT@EC service, receiving a notification only when the translation is complete, whereas the Wrapper uses the REST interface of MT@EC. The REST service allows real-time interaction with the translation system, evaluation of performance, exploiting of the MT@EC parallelism of translation resources and optimization the workload; above all, it allows the user to perform a real-time QC of the internationalization on progress.

The contribution of the Wrapper is to produce a more versatile corpus in comparison to its major competitor, the EDP. In general, the Wrapper achieves its specific goal of fully automating the internationalization process; in all the other systems described in Section 2.2, the automation is only partial, in that it requires human intervention at some point, for process setting or data import/export between different software and systems. To accomplish this goal, the Wrapper has a number of process control features, and solutions for optimizing performance, resulting from its specific design and architectural choice.

5. CONCLUSIONS AND DIRECTIONS FOR FURTHER RESEARCH

In terms of the quality of the parallel corpus produced by the Wrapper, two aspects should be highlighted. The parallel corpus produced in this experimental phase internationalizes only a few of the core elements in ISO 19115, with which the functionality of the system is demonstrated. Internationalization of all the relevant elements for INSPIRE could be accomplished in an extension of this project. It also emerges from the literature review that the quality of translation would be likely to be improved if it were based on domain-specific translation memories. The MT@EC service can use domain-specific engines, although at present this feature is limited to users who have provided data to build specific engines. Improvements to the internationalization therefore require a close collaboration with the MT@EC project, and should aim at the construction of domain-specific engines for geographic domains.

The MT@EC-Wrapper is original in that it proposes the complete automation of the internationalization process of XML corpora. This prototype illustrates the potential of combining modern open-source tools in a document automation application that can be applied in both industry and research. The microservice architecture adopted makes the system easily expandable, and the use of asynchronous communications makes it inherently reliable in processing documents from several sources with frequent anomalies in their structure. This significantly simplifies the development, as it integrates the web server's ability to manage competing processes. This distributed and modular architecture can be easily extended to handle several projects within an organization, and to support further text processing such as text mining.

6. ACKNOWLEDGEMENT

This work as carried out as a technical support to the Joint Research Centre (JRC) of the European Commission. As such, the views expressed are purely those of the author and may not in any circumstances be regarded as stating an official position of the European Commission.

REFERENCES

Ahronheim, J.R. (1998). Descriptive metadata: Emerging standards. The Journal of Academic Librarianship 25(5): 395-403.

Anastasiou, D. (2010). Survey on the Use of XLIFF in Localisation Industry and Academia. International Conference on Language Resources and Evaluation (LREC), Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC), pp. 50-53.

Aswani, N. (2012). Designing a General Framework for Text Alignment: Case Studies with Two South Asian Languages, Doctoral dissertation, University of Sheffield.

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R. and Ives, Z. (2007). DBpedia: a nucleus for a web of open data. In: Aberer, K. et al. (Ed.) The Semantic Web. Lecture Notes in Computer Science, vol. 4825. Springer, Berlin, Heidelberg, pp. 722-735.

Esselink, B. and O'Brien, S. (2000). A Practical Guide to Localization. John Benjamins Publishing Company.

Fernandes, S. (2012). Using XML Schemas in Parallel Corpora. Proceedings of Doctoral Symposium in Informatics Engineering (DSIE), pp. 335-346.

Gale, W.A. and Church, K.W. (2014). A program for aligning sentences in bilingual corpora. Computational Linguistics 19(1): 75-102.

Gayo, J.E.L, Kontokostas, D., Auer, S. (2013). Multilingual linked open data patterns. Semantic Web Journal.

INSPIRE (2009). INSPIRE Metadata Implementing Rules: Technical Guidelines based on EN ISO 19115 and EN ISO 19119. European Commission.

INSPIRE (2013). INSPIRE Generic Conceptual Model. European Commission.

Ishida, R. and Miller, S.K. (2016), Localization vs. Internationalization. W3C at https://www.w3.org/International/questions/qa-i18n.

ISO (2012). ISO 24616:2012 Language resources management - Multilingual information framework. Reference Number: ISO 24616:2012(E). International Organization for Standardization, Geneva, Switzerland

ISO (2014). ISO 19115:2003 Geographic information - Metadata. Reference Number: ISO 19115:2003(E). International Organization for Standardization, Geneva, Switzerland.

ISO (2015). ISO/TS 19103:2005 Geographic information - Conceptual schema language. Reference Number: ISO 19103:2005(E). International Organization for Standardization, Geneva, Switzerland

Kresse, W. and Fadaie, K. (2010). ISO Standards for Geographic Information. Springer Science and Business Media. Springer.

Laurent, R. (2015). Standards for language resources in ISO - Looking back at 13 fruitful years. Die Terminologiefachzeitschrift.

LIND (2016). Language Industry Survey - Expectations and Concerns of the European Language Industry. European Commission.

LISA (2003). Translation Memory eXchange, Localization Industry Standards Association at https://www.gala-global.org/tmx-14b.

LTC (2009). Study on the Size of the Language Industry in the EU. European Commission.

OASIS (2007). XLIFF Version 1.2. Organization for the Advancement of Structured Information Standards at http://docs.oasis-open.org/xliff/v1.2/cs02/xliff-core.html. 2007.

Pawar, P., Ardhapurkar, P., Jain, P., Lele, A., Kumar, A. and Darbari, H. (2015). XLIFF: complementary for a complete localization of machine translation among divergent language families. Proceedings of the Fifth International Conference on Communication Systems and Network Technologies (CSNT), pp. 1260-1264.

Sasaki, F. (2009). Markup Languages and Internationalization. In: Witt, A. and Metzing, D. (Ed.) Linguistic Modeling of Information and Markup Languages, Springer, pp. 67-80.

Schaler, R. (2010). Localization and translation. In: Gambier, Y. and Van Doorslaer, L. (Ed.) Handbook of translation studies. John Benjamins Publishing Company, pp. 209-214.

Simoes, A. (2004). Parallel corpora word alignment and applications. Thesis, Braga, Escola de Engenharia, Universidade do Minho.

Sourceforge (2009). XML Benchmark Results 10.10.2009. Sourceforge at http://xmlbench.sourceforge.net/results/benchmark/index.html.

Steinberger, R., Pouliquen, B., Widiger, A., Ignat, C., Erjavec, T., Tufis, D. and Varga,D. (2006). The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. Proceedings of the International Conference on Language Resources and Evaluation (LREC), pp. 2142-2147.

Steinberger, R., Ebrahim, M., Poulis, A., Carrasco-Benitez, M., Schlüter, P., Przybyszewski, M. and Gilbro,S. (2014). An overview of the European Union's highly multilingual parallel corpora. Language Resources and Evaluation 48(4): 679-707.

Steinberger, R., Eisele, A., Klocek, S., Pilos, S. and Schlüter, P. (2012). DGT-TM: A Freely Available Translation Memory in 22 Languages. Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC), pp. 454-459.

The Unicode Consortium (2006). The Unicode Standard, Version 5.09. Addison Wesley.

Valli, P. (2012). Translation practice at the EU institutions: focus on a concordancing tool. Rivista Internazionale di Tecnica della Traduzione 14: 95-109.

Vanhoutte, E. (2004). An Introduction to the TEI and the TEI Consortium, Literary and Linguistic Computing, 19(1): 9-16.

Varga, D., Halácsy, P., Kornai, A., Nagy, V., Németh, L. and Trón, V. (2007). Parallel corpora for medium density languages. Amsterdam Studies in the Theory and History of Linguistic Science 4: 292-297*.*

Wolff, E. (2016). Microservices: Flexible Software Architecture. Addison Wesley.

Wright, S.E. (2013). Standardization in Human Language Technology. In: Chapelle, C.A. (Ed.) Encyclopedia of Applied Linguistics. Wiley-Blackwell.